



TUGAS AKHIR - SM141501

**PENGUKURAN KECEPATAN KENDARAAN
MENGUNAKAN *OPTICAL FLOW***

**ACHMAD ROMLI
NRP. 1213 100 105**

Dosen Pembimbing :
Dr. Dwi Ratna Sulistyaningrum, S.Si, MT

DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - SM141501

**PENGUKURAN KECEPATAN KENDARAAN
MENGUNAKAN *OPTICAL FLOW***

**ACHMAD ROMLI
NRP 1213 100 105**

**Pembimbing:
Dr. Dwi Ratna Sulistyaningrum, S.Si, MT**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

Halaman ini sengaja dikosongkan



FINAL PROJECT - SM141501

VEHICLE SPEED MEASUREMENT USING OPTICAL FLOW

ACHMAD ROMLI
NRP 1213 100 105

Supervisor:
Dr. Dwi Ratna Sulistyaningrum, S.Si, MT

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Sciences
Sepuluh Nopember Institute of Technology
Surabaya 2017

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN
PENGUKURAN KECEPATAN KENDARAAN
MENGUNAKAN *OPTICAL FLOW*

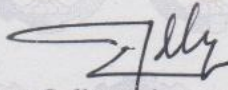
VEHICLE SPEED MEASUREMENT USING
OPTICAL FLOW

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
ACHMAD ROMLI
NRP. 1213 100 105

Menyetujui,
Dosen Pembimbing,



Dr. Dwi Ratna Sulistyaningrum, S.Si, MT
NIP. 19690405 199403 2 003

Mengetahui,

Kepala Departemen Matematika
MIPA ITS



Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003
Surabaya, Agustus 2017

Halaman ini sengaja dikosongkan

PENGUKURAN KECEPATAN KENDARAAN MENGUNAKAN *OPTICAL FLOW*

Nama Mahasiswa : Achmad Romli
NRP : 1213 100 105
Departemen : Matematika FMIPA-ITS
Pembimbing : Dr. Dwi Ratna Sulistyaningrum,
S.Si, MT

Abstrak

Berdasarkan Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan Pasal 287 ayat 5, setiap pengendara yang melanggar aturan batas kecepatan paling tinggi atau paling rendah akan dipidana. Upaya-upaya telah dilakukan untuk mengamati kecepatan kendaraan. Diantaranya pengukuran kecepatan kendaraan berbasis pengolahan citra digital. Pada Tugas Akhir ini pengukuran kecepatan kendaraannya akan memanfaatkan metode *optical flow*. Objek dari penelitian ini adalah video rekaman kendaraan di jalan raya. Pengukuran kecepatan kendaraan dilakukan beberapa langkah, yaitu; menentukan skala perbandingan, memilih area, memproses area terpilih dengan metode *Improved Three Frame Difference*, menghitung nilai *optical flow* dari objek pada area, dan menghitung kecepatan dari objek yang diperoleh dari nilai *optical flow* objek dikalikan dengan skala perbandingan. Dari uji coba yang sudah dilakukan didapatkan hasil bahwa metode *optical flow* Horn-Schunck mampu mendeteksi kecepatan kendaraan dengan tingkat akurasi paling tinggi 89,37%.

Kata kunci: *Optical flow, Improved Three Frame Difference, Speed Detection.*

Halaman ini sengaja dikosongkan

VEHICLE SPEED MEASUREMENT USING OPTICAL FLOW

Name : Achmad Romli
NRP : 1213 100 105
Department : Mathematics FMIPA-ITS
Supervisor : Dr. Dwi Ratna Sulistyaningrum, S.Si, MT

Abstract

Based on Law Number 22 Year 2009 on Road Traffic and Transportation Clause 287 verse 5, every rider who violates the rules of the maximum speed limit or the lowest shall be punished. Attempts have been made to observe the speed of the vehicle. Among the measurement of vehicle speed based digital image processing. In this Final Project measurement of vehicle speed will utilize optical flow method. The object of this research is the video recording of vehicles on the highway. Vehicle speed measurements performed several steps, namely; Determine the comparison scale, select the area, process the selected area by the Improved Three Frame Difference method, calculate the optical flow value of the object in the area, and calculate the velocity of the object obtained from the optical flow value of the object multiplied by the comparison scale. From the experiments that have been done the results obtained that the method of optical flow Horn-Schunck able to detect the speed of the vehicle with the highest accuracy of 89,37%.

Keywords: *Optical flow, Improved Three Frame Difference, Speed Detection.*

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Assalamualaikum Warohmatullohi Wabarokatuh.

Syukur alhamdulillah atas kehadiran Allah SWT yang telah melimpahkan segala nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

“PENGUKURAN KECEPATAN KENDARAAN MENGUNAKAN *OPTICAL FLOW*”

sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika FMIPA Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas Akhir ini dapat terselesaikan dengan baik berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih dan penghargaan kepada:

1. Ibu Dr. Dwi Ratna Sulistyaningrum, S.Si., MT sebagai dosen pembimbing yang telah memberikan motivasi dan pengarahan dalam penyelesaian Tugas Akhir ini.
2. Bapak Dr. Budi Setiyono, S.Si, MT, Bapak Drs. Soetrisno, MI.Komp, Bapak Drs. Kamiran, M.Si, dan Bapak Drs. Iis Herisman, M.Si. sebagai dosen penguji yang telah memberikan banyak kritik dan saran dalam penyelesaian Tugas Akhir ini.
3. Bapak Dr. Imam Mukhlash, S.Si., MT sebagai Kepala Departemen Matematika FMIPA-ITS.
4. Bapak Dr. Didik Khusnul Arif, S.Si., M.Si. sebagai Kaprodi Sarjana Matematika FMIPA-ITS dan Bapak Drs. Iis Herisman, M.Si. sebagai Sekprodi Sarjana Matematika FMIPA-ITS.
5. Jajaran dosen dan Staf Departemen Matematika ITS.
6. Bapak Dr. Mahmud Yunus, M.Si sebagai dosen wali selama penulis kuliah di Departemen Matematika FMIPA-ITS.

7. Ibu, Ayah, Mbak, Mas dan keluarga besar saya yang telah banyak mendukung dan memberi semangat saya dalam menjalani masa perkuliahan.
8. Teman saya dan teman-teman mahasiswa, faisal, agus, ardi, wawan, ivan, firdo, mbak farah, mas habib, mas danang, gery, faul, bayu, kunur, haidar, firdo, satria, bhara, daus, rozi, fadhlan, adit, oby, alwi, ariel, azzam, zani, asna, rifki, angga, ikun, yoga, anshor, jonbon, uzu, yos, prima, dan seluruh mahasiswa Matematika 2013 serta keluarga STI-48 LAMBDA (yang tidak dapat saya sebutkan satu persatu) yang telah banyak mendukung baik dari doa maupun tindakan yang telah memberi semangat saya saat menjalani perkuliahan.
9. Semua pihak yang telah memberikan dukungan dan ilmu kepada penulis dalam penyelesaian Tugas Akhir ini.

Apabila dalam penulisan Tugas Akhir ini ada kekurangan, kami mohon kritik dan saran demi penyempurnaan Tugas Akhir ini di masa yang akan datang. Semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak. Amin.

Surabaya, 1 Agustus 2017

Penulis

DAFTAR ISI

	Hal
TUGAS AKHIR - SM141501.....	i
Abstrak.....	vii
<i>Abstract</i>	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan.....	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	3
BAB II.....	7
TINJAUAN PUSTAKA	7
2.1 Penelitian Terdahulu.....	7
2.2 Metode <i>Improved three-frame difference</i>	8
2.2.1 Metode <i>Interframe difference</i>	8
2.2.2 Metode <i>three-frame difference</i> yang ditingkatkan. .	9
2.3 Persamaan dasar dari <i>optical flow</i> dan algoritma yang ditingkatkan	13

	Hal
2.3.1 Persamaan dasar dari <i>optical flow</i>	13
2.3.2 Algoritma <i>gray constraint optical flow</i>	14
2.3.3 Persamaan <i>motion constraint</i> Horn dan Schunck..	16
2.4 Pengukuran kecepatan kendaraan.....	18
BAB III	21
METODE PENELITIAN.....	21
3.1 Studi Literatur.....	21
3.2 Analisis dan Desain Sistem	21
3.3 Implementasi Program.....	22
3.4 Uji Coba dan Evaluasi Program	22
3.5 Penarikan Kesimpulan dan Saran	23
3.6 Penulisan Laporan Tugas Akhir	23
BAB IV	25
PERANCANGAN DAN IMPLEMENTASI SISTEM.....	25
4.1 Analisis Sistem	25
4.1.1 Analisa Sistem Perangkat Lunak.....	25
4.1.2 Analisis Kebutuhan Sistem	29
4.2 Perancangan Sistem.....	30
4.2.1 Perancangan Data Sistem	30
4.2.2 Perancangan Proses	32
4.2.3 Perancangan Antar Muka Sistem	36
4.3 Implementasi Sistem.....	38
4.3.1 Implementasi <i>Input</i> Video	39

	Hal
4.3.2 Implementasi Pemilihan area ROI	40
4.3.3 Implementasi Proses <i>Improved Three Frame Difference</i>	41
4.3.4 Implementasi Pendeteksian Kecepatan Kendaraan	42
BAB V	45
UJI COBA DAN PEMBAHASAN.....	45
5.1 Data Uji Coba	45
5.2 Uji Coba Pendeteksian Kecepatan Kendaraan Bergerak	46
5.2.1 Rekaman Video dengan Kecepatan 20 Km/Jam ...	46
5.2.2 Rekaman Video dengan Kecepatan 30 Km/Jam ...	47
5.2.3 Rekaman Video dengan Kecepatan 40 Km/Jam ...	47
5.2.4 Rekaman Video dengan Kecepatan 50 Km/Jam ...	48
5.2.5 Rekaman Video dengan Kecepatan 30 dan 40 Km/Jam	48
5.2.6 Pembahasan Uji Coba Pendeteksian Kecepatan Kendaraan.....	49
BAB VI.....	51
PENUTUP	51
6.1 Kesimpulan.....	51
6.2 Saran	51
DAFTAR PUSTAKA	53
LAMPIRAN A.....	57
LAMPIRAN B	61

	Hal
BIODATA PENULIS	63

DAFTAR GAMBAR

	Hal
Gambar 2.1 Efek dari <i>overlap</i> -nya <i>three-frame difference</i>	9
Gambar 2.2 Efek dari <i>three-frame difference</i> yang ditingkatkan.	11
Gambar 2.3 Citra asli: (a) citra ke-68, (b) citra ke-70, dan (c) citra ke-72.	12
Gambar 2.4 Efek dari metode <i>difference</i> : (a) citra metode <i>two-frame difference</i> , (b) citra metode <i>three-frame difference</i> , dan (c) citra metode <i>three-frame difference</i> yang ditingkatkan.	12
Gambar 2.5 Diagram skema instalasi.	19
Gambar 2.6 <i>Region of interest</i>	20
Gambar 3.1 Blok diagram sistem pengukuran kecepatan kendaraan.	24
Gambar 4.1 DFD level 0 program estimasi kecepatan kendaraan.	26
Gambar 4.2 DFD level 1 program estimasi kecepatan kendaraan.	26
Gambar 4.3 DFD Level 2 dari proses segmentasi.....	26
Gambar 4.4 DFD Level 2 dari proses penghitungan kecepatan.	27
Gambar 4.5 Proses <i>scanning</i> video.	28
Gambar 4.6 Citra dengan ROI (daerah didalam kotak merah).	28
Gambar 4.7 Citra hasil <i>Background Subtraction</i> dengan metode <i>Improved Three Frame Difference</i> . (a) citra asli, (b) citra <i>foreground</i>	29
Gambar 4.8 <i>Business Rule</i> sistem perangkat lunak.....	30
Gambar 4.9 <i>Layout</i> untuk pengambilan video.	31
Gambar 4.10 Diagram alir sistem perangkat lunak.....	34

	Hal
Gambar 4.11 Diagram alir <i>Improved Three Frame Difference</i>	35
Gambar 4.12 Diagram alir deteksi kecepatan.	37
Gambar 4.13 Antar muka halaman utama.	38
Gambar 4.14 Antar muka halaman detail.	39
Gambar 4.15 Antar muka <i>input</i> video.....	40
Gambar 4.16 Kotak merah pada <i>frame</i> tersebut adalah area ROI yang dipilih.....	41
Gambar 4.17 Antar muka aplikasi saat pendeteksian kecepatan kendaraan.	44

DAFTAR TABEL

	Hal
Tabel 4.1 Tabel data proses.	32
Tabel 5.1 Tabel Data Video yang Digunakan.....	45
Tabel 5.2 Tabel Pendeteksian Kecepatan Kendaraan dengan Kecepatan 20 Km/Jam.	47
Tabel 5.3 Tabel Pendeteksian Kecepatan Kendaraan dengan Kecepatan 30 Km/Jam.	47
Tabel 5.4 Tabel Pendeteksian Kecepatan Kendaraan dengan Kecepatan 40 Km/Jam.	48
Tabel 5.5 Tabel Pendeteksian Kecepatan Kendaraan dengan Kecepatan 50 Km/Jam.	48
Tabel 5.6 Tabel Pendeteksian Kecepatan Kendaraan dengan Kecepatan 30 dan 40 Km/Jam.....	49
Tabel 5.7 Tabel Prosentase Keberhasilan Video Uji Coba...	50

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Bab ini membahas latar belakang yang mendasari penulisan Tugas Akhir. Didalamnya mencakup identifikasi permasalahan pada topik Tugas Akhir. Uraian ini bersifat umum yang menjelaskan secara ringkas hal-hal yang dilakukan pada penyelesaian Tugas Akhir. Informasi yang telah diperoleh tersebut kemudian dirumuskan menjadi permasalahan, kemudian diberikan asumsi-asumsi dan batasan-batasan untuk membatasi pembahasan pada Tugas Akhir ini.

1.1 Latar Belakang

Berdasarkan Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, yang disahkan DPR pada 22 Juni 2009. Salah satu bunyi pasal tilang untuk kendaraan bermotor terhadap pelanggaran lalu lintas adalah setiap pengendara yang melanggar aturan batas kecepatan paling tinggi atau paling rendah dipidana dengan kurungan paling lama 2 bulan atau denda paling banyak Rp 500 ribu (Pasal 287 ayat 5).

Dengan adanya peraturan tersebut diharapkan mengurangi jumlah yang terjadi pelanggaran. Namun pada kenyataannya masih saja kerap terjadi kasus-kasus pelanggaran lalu lintas seperti kebut-kebutan di jalan atau balap liar. Hal tersebut terjadi karena minimnya pengawasan dari aparat penegak hukum. Sehingga peningkatan pengawasan sangat perlu dilakukan untuk mencegah hal-hal yang tidak diinginkan.

Upaya-upaya telah dilakukan untuk mengamati kecepatan kendaraan. Dan kebanyakan masih menggunakan metode *Laser Speed Measurement* dan *Radar Speed Measurement* untuk mengurangi pelanggaran lalu lintas batas kecepatan kendaraan. Metode *Laser Speed Measurement* masih memiliki kekurangan yakni performansinya berkurang ketika suhu ekstrim [1-2]. Dan Metode *Radar Speed Measurement*

mengharuskan sudut yang benar-benar pas saat pemasangannya, yang mana memiliki keterbatasan dalam penerapannya [3]. Selain itu, keduanya membutuhkan biaya yang besar untuk membeli alatnya. Sehingga untuk alternatif lain, pengawasan lalu lintas saat ini sudah memanfaatkan kamera CCTV yaitu *road traffic monitoring* (RTMC). Pemasangan kamera pada ruas-ruas jalan saat ini sudah banyak dilakukan. Selain itu penelitian tentang *intelligent transportation system* (ITS) juga memanfaatkan pengawasan melalui kamera. Sejalan dengan hal tersebut, maka hasil rekaman CCTV yang mana berupa video dapat dimanfaatkan untuk mengukur kecepatan kendaraan pengguna jalan.

Pengukuran kecepatan kendaraan berbasis pengolahan citra digital telah banyak dilakukan. Metode yang umum digunakan untuk pengukuran kecepatan kendaraan dengan citra video diantaranya *corner detection*, *texture analysis*, *video tracking*, dan seterusnya [4-9]. Oleh karena itu, pada Tugas Akhir ini akan dibahas pengukuran kecepatan kendaraan yang berbasis pengolahan video digital. Dalam pengukuran kecepatan kendaraannya akan memanfaatkan metode *optical flow*. Nilai *optical flow* adalah kecepatan (pixels/s) dari kendaraan dalam citra. Metode ini dapat mengurangi penghitungan dengan hanya menghitung nilai *optical flow* kontur target bergerak [10].

1.2 Rumusan Masalah

Rumusan masalah dari Tugas Akhir ini antara lain:

1. Bagaimana melakukan estimasi kecepatan kendaraan menggunakan *optical flow* agar diperoleh hasil yang akurat?
2. Bagaimana kinerja *optical flow* dalam melakukan estimasi kecepatan kendaraan?

1.3 Batasan Masalah

Untuk memfokuskan penelitian Tugas Akhir ini, maka perlu diambil beberapa batasan masalah sebagai berikut:

1. Data yang digunakan adalah rekaman video arus lalu lintas kendaraan di jalan raya pada saat pagi dan sore hari.
2. Sudut pandang yang diambil ketika merekam adalah dari atas dengan jangkauan pandangan seluruh jalan arus satu arah.
3. Menggunakan kamera digital dengan resolusi 480x640 dan 720x1280 dengan *framerate* sekitar 30 *frame per second* (fps).
4. Hanya objek tunggal yang diamati pada program.

1.4 Tujuan

Tujuan dari Tugas Akhir ini antara lain:

1. Melakukan estimasi kecepatan kendaraan menggunakan *optical flow*.
2. Mengukur kinerja algoritma *optical flow* dalam melakukan estimasi kecepatan kendaraan.

1.5 Manfaat

Manfaat yang diperoleh dari Tugas Akhir ini adalah sistem yang telah dibangun dapat memberikan informasi tentang kecepatan kendaraan yang melaju di jalan sehingga dapat dijadikan alat bukti bagi aparat penegak hukum untuk pelanggaran lalu lintas.

1.6 Sistematika Penulisan

Penulisan Tugas Akhir ini disusun dalam lima bab, yaitu:

1. BAB I PENDAHULUAN

Bab ini berisi tentang gambaran umum dari penulisan Tugas Akhir yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Dalam bab ini diuraikan mengenai penelitian yang terkait dalam pengukuran kecepatan kendaraan, kemudian diuraikan

metode pengolahan citra yakni *Improved Three Frame Difference* begitu pula algoritmanya. Metode tersebut merupakan pengembangan dari metode *Three Frame Difference*. Kemudian dijelaskan mengenai persamaan dasar *optical flow*, kemudian dijelaskan mengenai algoritma *optical flow* Horn-Schunck yang juga merupakan pengembangan dari *optical flow* sendiri. Yang terakhir dijelaskan bagaimana mendapatkan kecepatan kendaraan dari algoritma yang akan digunakan.

3. BAB III METODE PENELITIAN

Dalam bab ini diuraikan langkah-langkah sistematis yang dilakukan dalam proses pengerjaan Tugas Akhir. Kegiatan penelitian dalam Tugas Akhir ini terdiri atas: studi literatur, analisis dan desain system, implementasi program, evaluasi dan desain program, penarikan kesimpulan dan saran, dan yang terakhir penulisan laporan Tugas Akhir.

4. BAB IV PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dibahas mengenai perancangan dan implementasi sistem dimulai dari pembahasan proses pengambilan data masukan hingga penjelasan mengenai cara untuk mendapatkan data keluaran yang sesuai dengan tujuan dari penelitian Tugas Akhir ini. Selain hal yang telah disebutkan di atas, perancangan sistem juga meliputi perancangan antar muka (*user interface*). Hasil dari analisis dan perancangan sistem dilanjutkan dengan implementasi sistem.

5. BAB V UJI COBA DAN PEMBAHASAN

Bab ini menjelaskan mengenai pengujian program dan pembahasan dari hasil uji coba. Pengujian yang dilakukan disini adalah pengujian program dengan *input* video rekaman kendaraan bergerak di jalan raya tepatnya di jalan Kedung Cowek, Jembatan Surabaya, dan Wonokromo, Surabaya.

Pengambilan video dilakukan pada waktu pagi dan sore hari. Durasi rekaman yang digunakan pada pengujian ini adalah 1 sampai 6 detik dengan objek yang diamati dalam video tersebut adalah sepeda motor.

6. BAB VI PENUTUP

Dalam bab ini berisi kesimpulan akhir yang diperoleh dari Tugas Akhir serta saran untuk pengembangan penelitian selanjutnya.

Halaman ini sengaja dikosongkan

BAB II

TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai dasar teori yang digunakan dalam penyusunan Tugas Akhir ini. Dasar teori yang digunakan meliputi dalam Penelitian terdahulu, penjelasan tentang *frame difference*, penjelasan tentang *three-frame difference* dan algoritmanya, penjelasan metode *three-frame difference* yang ditingkatkan dan gambaran dari algoritmanya. Kemudian dijelaskan mengenai persamaan dasar *optical flow*, penjelasan mengenai algoritma *optical flow* Horn-Schunck. Yang terakhir dijelaskan bagaimana mendapatkan kecepatan kendaraan dari algoritma yang akan digunakan.

2.1 Penelitian Terdahulu

Beberapa penelitian dengan tema yang terkait dengan Tugas Akhir ini diantaranya adalah penelitian yang dilakukan oleh A. Fernandez-Caballero dkk. yang berjudul “*Road-Traffic Monitoring by Knowledge-Driven Static and Dynamic Image Analysis*”[4], oleh Y. Li dkk. yang berjudul “*Vehicle Speed Measurement Based on Video Images*”[5], oleh R. Canals dkk. yang berjudul “*A Biprocessor-Oriented Vision-Based Target Tracking System*”[6], oleh R. Cucchiara dkk. yang berjudul “*Detecting Moving Objects, Ghosts, and Shadows in Video Streams*”[7], oleh T. Schoepflin dkk. yang berjudul “*Dynamic Camera Calibration of Roadside Traffic Management Cameras for Vehicle Speed Estimation*”[8], oleh Z.-D. Zhang dan C.-w. Zhang yang berjudul “*Method to Improve The Accuracy of Video-Based Vehicle Speed Measurement*”[9]. Mereka semua menggunakan metode-metode yang umum dalam mengukur kecepatan kendaraan dengan citra video yakni *corner detection*, *texture analysis*, *video tracking*, dan seterusnya. Metode-metode ini dapat mencari dengan baik lokasi yang cocok dari kendaraan dalam dua *frame* gambar berturut-turut, tetapi jumlah perhitungan yang besar tidak bisa

memenuhi persyaratan dari pengukuran kecepatan yang sebenarnya.

2.2 Metode *Improved three-frame difference*

2.2.1 Metode *Interframe difference*

Metode *interframe difference* secara luas digunakan pada deteksi target bergerak dengan kerumitan penghitungan yang rendah. Dalam banyak kasus, kinerjanya baik. Ketika target bergerak cepat, kontur target yang dihasilkan oleh dua *frame* citra video menghasilkan *overlap*. Jadi, metode *three-frame difference* diusulkan [11].

Prinsip dari metode *three-frame difference* ditunjukkan dalam persamaan (2.1) - (2.3):

$$p_1(x, y) = \begin{cases} 255, & \text{jika } |F_m(x, y) - F_{m-1}(x, y)| > T \\ 0, & \text{jika } |F_m(x, y) - F_{m-1}(x, y)| \leq T \end{cases} \quad (2.1)$$

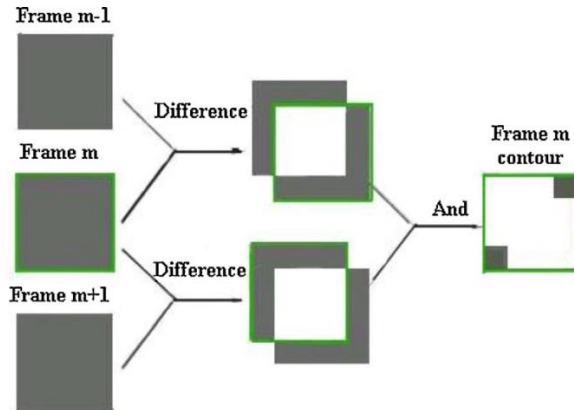
$$p_2(x, y) = \begin{cases} 255, & \text{jika } |F_{m+1}(x, y) - F_m(x, y)| > T \\ 0, & \text{jika } |F_{m+1}(x, y) - F_m(x, y)| \leq T \end{cases} \quad (2.2)$$

$$\begin{aligned} p(x, y) &= p_1(x, y) \cap p_2(x, y) \\ &= \begin{cases} 255, & \text{jika } (p_1(x, y) \& p_2(x, y) = 255) \\ 0, & \text{yang lain} \end{cases} \end{aligned} \quad (2.3)$$

Dalam persamaan (2.1) dan (2.2), $F_{m-1}(x, y)$, $F_m(x, y)$, $F_{m+1}(x, y)$ menotasikan *gray value* dari $m - 1$ frame citra, m frame citra, dan $m + 1$ frame citra pada *spatial pixel point* (x, y) . T adalah *threshold value*. $p(x, y)$ adalah *gray value* dari *spatial pixel point* (x, y) dari citra target.

Meskipun metode *three-frame difference* dapat memperoleh kontur target bergerak, kontur yang dideteksi ter-*overlap* sebagian, yang mana dibandingkan dengan kontur target sebenarnya. Kinerja dari metode ditunjukkan dalam Gambar

2.1 Kontur hijau adalah kontur target dan daerah-daerah abu-abu adalah kawasan *overlap* dalam *m* kontur *frame*.



Gambar 2.1 Efek dari *overlap*-nya *three-frame difference*.

2.2.2 Metode *three-frame difference* yang ditingkatkan.

Hasil dari metode *two-frame difference* menghasilkan *overlap*, dan algoritma *three-frame difference* juga menghasilkan *overlap* sebagian. Algoritma itu dapat mendeteksi kontur target dengan tepat, dan dapat memperoleh informasi kontur melalui operasi dari “*difference*”, “*dilate*”, “*and*”, “*xor*” dalam tiga frame berturut-turut.

Langkah-langkah algoritma *three-frame difference* yang ditingkatkan adalah sebagai berikut:

- 1) Perbedaan citra p_1 diperoleh dengan *frame* pertama dan *frame* citra kedua melalui operasi “*difference*”.

$$p_1(x, y) = \begin{cases} 255, & \text{jika } |F_m(x, y) - F_{m-1}(x, y)| > T \\ 0, & \text{jika } |F_m(x, y) - F_{m-1}(x, y)| \leq T \end{cases}$$

- 2) Perbedaan citra p_2 diperoleh dengan *frame* kedua dan *frame* citra ketiga melalui operasi “*difference*”.

$$p_2(x, y) = \begin{cases} 255, & \text{jika } |F_{m+1}(x, y) - F_m(x, y)| > T \\ 0, & \text{jika } |F_{m+1}(x, y) - F_m(x, y)| \leq T \end{cases}$$

- 3) Citra baru p_3 diturunkan dari p_1 dan p_2 melalui operasi “*and*”.

$$\begin{aligned} p_3(x, y) &= p_1(x, y) \cap p_2(x, y) \\ &= \begin{cases} 255, & \text{jika } (p_1(x, y) \& p_2(x, y) = 255) \\ 0, & \text{yang lain} \end{cases} \end{aligned}$$

- 4) Secara berturut citra baru p_4 dan p_5 atau citra dilatasi diperoleh dari penambahan template persegi 3×3 (*SE*) untuk melebarkan secara berturut citra p_1 dan p_2 .

$$p_4(x, y) = p_1(x, y) \oplus SE$$

$$p_5(x, y) = p_2(x, y) \oplus SE$$

- 5) Citra baru p_6 diturunkan dari citra dilatasi p_4 dan p_5 melalui operasi “*and*”.

$$\begin{aligned} p_6(x, y) &= p_4(x, y) \cap p_5(x, y) \\ &= \begin{cases} 255, & \text{jika } (p_4(x, y) \& p_5(x, y) = 255) \\ 0, & \text{yang lain} \end{cases} \end{aligned}$$

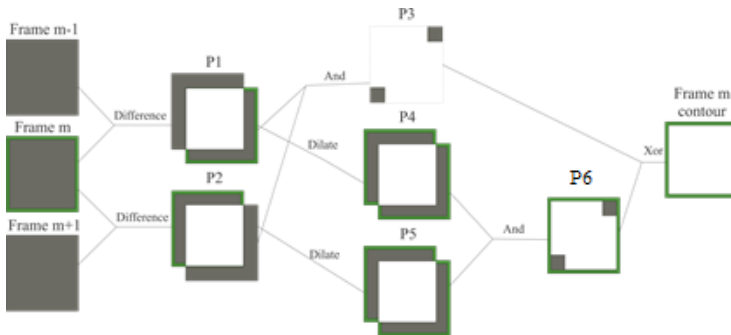
- 6) Hasil citra p diturunkan dari p_3 dan p_6 melalui operasi “*xor*”.

$$\begin{aligned} p(x, y) &= p_3(x, y) \otimes p_6(x, y) \\ &= \begin{cases} 255, & \text{jika } (p_3(x, y) \neq p_6(x, y)) \\ 0, & \text{yang lain} \end{cases} \end{aligned}$$

Itu juga diketahui bahwa *threshold* yang global T untuk binarisasi citra dapat mengakibatkan distorsi pada citra yang asli yang intensitasnya bervariasi secara bertahap karena pencahayaan. Oleh karena itu, diadopsi metode *local adaptive threshold segmentation* untuk memekstrak target dari *background*. Global adaptive threshold T digunakan untuk melakukan segmentasi awal. Nilai *threshold* T untuk seluruh citra ditentukan oleh perbedaan antara mean dan standard deviasi dari seluruh citra.

Nilai *threshold* $T = \text{mean} - \text{std}$.

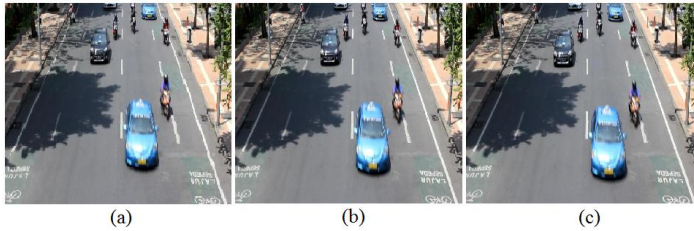
Efek dari algoritma *three-frame difference* yang ditingkatkan ditunjukkan pada Gambar 2.2. Dari gambar, kontur presisi dari target dapat diperoleh dari metode. Jadi, algoritma tersebut mempunyai hasil yang lebih baik daripada algoritma *three-frame difference*. Secara khusus, ketika kecepatan dari target bergerak lebih cepat, efeknya lebih jelas.



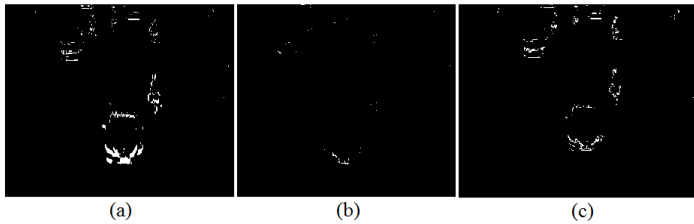
Gambar 2.2 Efek dari *three-frame difference* yang ditingkatkan.

Untuk mengilustrasikan kinerja yang unggul metode yang ditingkatkan, percobaan dilakukan dengan tiga jenis metode *frame difference*. Gambar 2.3. Menunjukkan citra asli, dan berukuran 480×720 pixel. Nilai *threshold* yang dihitung T

adalah 64,23 dari citra. Gambar 2.4(a) menunjukkan kinerja dari metode *two-frame difference*, Gambar 2.4(b) menunjukkan kinerja metode *three-frame difference* dan Gambar 2.4(c) menunjukkan kinerja metode yang diusulkan.



Gambar 2.3 Citra asli: (a) citra ke-68, (b) citra ke-70, dan (c) citra ke-72.



Gambar 2.4 Efek dari metode *difference*: (a) citra metode *two-frame difference*, (b) citra metode *three-frame difference*, dan (c) citra metode *three-frame difference* yang ditingkatkan.

Dari Gambar 2.4, dapat dilihat bahwa: Metode *two-frame difference* menghasilkan *overlap* secara jelas. Metode *three-frame difference* menghasilkan *overlap* pada sudut dari kontur mobil. Kontur yang dihasilkan dengan menggunakan algoritma yang diusulkan tepat, yang mana ditunjukkan pada Gambar 2.4(c). Jadi, metode *three-frame difference* yang ditingkatkan punya kinerja yang baik.

Untuk mendeteksi gerakan multi-target, algoritma yang diusulkan dapat digunakan sebagai partisi awal, dan

selanjutnya menggunakan metode *gray constraint optical flow* untuk menghitung kecepatan target bergerak.

2.3 Persamaan dasar dari *optical flow* dan algoritma yang ditingkatkan

Perubahan *gray value* yang diungkapkan oleh bidang *optical flow* dapat digunakan untuk mengamati informasi dari target bergerak dalam citra. Hal ini berisi informasi dari vektor kecepatan sesaatnya titik citra dari target. Berdasarkan pada vektor kecepatan, kecepatan dari target dapat dihitung.

2.3.1 Persamaan dasar dari *optical flow*

Optical flow pertama kali diusulkan oleh Gibson, dan yang mana adalah kecepatan bergerak dari target dalam *gray mode*. Perhitungan dari bidang *optical flow*-nya target bergerak berdasarkan pada dua asumsi: (1) *gray value* dari suatu objek bergerak tidak dirubah dalam waktu yang sangat singkat; (2) setiap pixel dari target punya kecepatan yang sama karena dari sifat kakunya target [12].

Misalkan bahwa titik P dalam koordinat citra pada (x, y) bergerak ke $(x + dx, y + dy)$ selama dt . *Gray value* adalah $I(x, y, t)$ pada (x, y) dan $I(x + dx, y + dy, t + dt)$ pada $(x + dx, y + dy)$. Karena dt pendek, *gray value* tidak diberubah oleh asumsi dan hal ini ditunjukkan pada persamaan (2.4). Juga, persamaan (2.4) disebut persamaan dasar *optical flow*.

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \quad (2.4)$$

Ruas kiri dari persamaan (2.4) diperluas dengan menggunakan rumus Taylor, dan ditunjukkan sebagai berikut.

$$\begin{aligned} I(x + dx, y + dy, t + dt) = I(x, y, t) &+ \frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} \\ &+ \frac{\partial I}{\partial t} \frac{dt}{dt} + o(\delta t^2) \end{aligned}$$

Setelah penyederhanaan dan elipsis batas kuadratik, persamaan dasar *optical flow* (2.4) disederhanakan ke persamaan (2.5).

$$\frac{\partial I}{\partial x} \frac{dx}{dt} + \frac{\partial I}{\partial y} \frac{dy}{dt} + \frac{\partial I}{\partial t} \frac{dt}{dt} = 0 \quad (2.5)$$

$u = dx/dt$ dan $v = dy/dt$ adalah kecepatan gerak dari pixel pada titik pixel spasial citra (x, y) dalam arah horizontal dan vertikal. Juga, (u, v) disebut bidang *optical flow*. Persamaan dasar *optical flow* dapat dinyatakan sebagai (2.6)

$$(I_x \ I_y)(u \ v)^T + I_t = 0 \quad (2.6)$$

dengan $I_x = \partial I / \partial x$, $I_y = \partial I / \partial y$, $I_t = \partial I / \partial t$.

2.3.2 Algoritma *gray constraint optical flow*

Dalam aplikasi praktis, asumsi bahwa *gray value* tidak berubah dalam waktu yang sangat singkat pada bidang *optical flow* tidak dapat dipenuhi karena alasan dari *block*, *multi-source*, *transparency*, dan seterusnya. Menurut *generalized dynamic image mode* (GDIM), *gray value* tidak konstan, tetapi diubah. Jadi, persamaan dasar *optical flow* (2.4) dan (2.5) tidak digunakan, sehingga diusulkan persamaan *gray constraint optical flow*, dan persamaan tersebut dapat dinyatakan sebagai berikut:

$$I(r + \Delta r) = M(r)I(r) + C(r) \quad (2.7)$$

dimana $I(r) = I(x, y, t)$, $M(r) = 1 + \Delta m(r)$, $\Delta m(r)$ adalah koefisien deviasi dan $C(r)$ adalah gangguan [10].

Persamaan *gray constraint optical flow* (2.7) ditunjukkan sebagai berikut:

$$I(r + \Delta r) = I(r) + \Delta M(r)I(r) + C(r) \quad (2.8)$$

Jika $\Delta M(r) = C(r) = 0$, persamaan (2.8) mendapati asumsi *gray value* yang tidak diubah dari persamaan dasar *optical flow*, dan itu sama dengan persamaan (2.4). Dalam aplikasi praktisnya, persamaan *gray constraint optical flow* (2.7) disesuaikan untuk situasi yang aktual. Juga, eror *gray constraint optical flow* ditunjukkan seperti dibawah:

$$\begin{aligned}\Delta I &= I(r + \Delta r) - I(r) = \Delta M(r)I(r) + C(r) \\ &= I_x u + I_y v + I_t\end{aligned}\quad (2.9)$$

Setelah deformasi, (2.9) dinyatakan sebagai berikut:

$$\Delta I = \Delta g + I_t \quad (2.10)$$

dengan $\Delta g = (I_x u + I_y v)$.

Dalam persamaan (2.10), Δg adalah komponen geometris (kecepatan dari objek yang bergerak), ΔI adalah komponen pencahayaan, dan rasio $Y = \Delta g / \Delta I$ dapat digunakan sebagai parameter dari *realtive strength* antara Δg dan ΔI . Jika rasionya besar, komponen yang mengestimasi kecepatan sangat akurat. Di sisi lain, estimasi kecepatan tidak akurat dan *affection of light* besar. Rasio ini penting untuk estimasi dari parameter gerak dan parameter struktural. Oleh karena itu, nilai Y dapat digunakan untuk membatasi distribusi bidang *optical flow*, yang mana dapat membuat algoritma *optical flow* bekerja dengan baik dalam berbagai cahaya yang menantang, kondisi lalu lintas, dan seterusnya.

Karena metode LK (Lucas-Kanade) boleh jadi tidak bekerja dengan baik pada bidang *dense flow*, metode HS (Horn dan Schunck) dapat mendeteksi sedikit gerakan dari objek dan menjadikan suatu bidang *flow* 100%. Dengan demikian, difokuskan pada metode HS untuk perhitungan bidang *optical flow* [13-14].

2.3.3 Persamaan *motion constraint* Horn dan Schunck

Untuk mendapatkan asumsi bahwa bidang *optical flow* yang disebabkan oleh objek yang sama-sama bergerak harus *smooth*, Horn dan Schunck mengusulkan kondisi *smoothness constraint*, yang bervariasi oleh integral dari komponen kecepatan persegi. Error *smoothness constraint* di nyatakan oleh persamaan (2.11). Ini panggilan untuk persamaan (2.11) untuk menjadi sekecil mungkin untuk tujuan *smoothness* [12,13].

$$E_s = \iint \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] dx dy \quad (2.11)$$

Dengan kata lain, asumsi bahwa *gray value* tidak diubah pada bidang *optical flow* tidak dapat dijumpai dengan banyak alasan. Sehingga, itu memerlukan bahwa error *gray constraint optical flow* yang diusulkan (2.9) sekecil mungkin, dan itu dinyatakan seperti (2.12).

$$E_c = \iint (\Delta I)^2 dx dy = \iint (I_x u + I_y v + I_t)^2 dx dy \quad (2.12)$$

Untuk mendapatkan error *optical flow* yang diusulkan dan error *smoothness constraint*, persamaan (2.13) dibuat untuk mencari solusi dari *optical flow* (u, v) .

$$E = E_c + \lambda E_s$$

$$= \iint \left\{ (I_x u + I_y v + I_t)^2 + \lambda \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] \right\} dx dy \quad (2.13)$$

λ adalah parameter yang menentukan beban antara dua error (error deviasi dari *smoothness* E_s dan error *optical flow* E_c). Ketika *gray value* citra dari pengukuran akurat, dan λ dapat dibuat lebih besar, yang mana memberikan penekanan lebih

besar pada E_c . Sebaliknya, jika citra mengandung banyak *noise*, λ adalah nilai kecil yang diinginkan (kurang dari 1).

Persamaan Euler yang sesuai dari (2.13) dinyatakan sebagai berikut:

$$\begin{aligned} \min[E_c + \lambda E_s] &= \\ &= \min \left[\iint \left\{ (I_x u + I_y v + I_t)^2 + \lambda \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] \right\} dx dy \right] \\ I_x(I_x u^{(n+1)} + I_y v^{(n+1)} + I_t) &= -\lambda \nabla^2 u \\ I_y(I_x u^{(n+1)} + I_y v^{(n+1)} + I_t) &= -\lambda \nabla^2 v \end{aligned} \quad (2.14)$$

Dimana ∇^2 adalah operator Laplace dan n menyatakan jumlah iterasi. Misalkan $\nabla^2 u = u^{(n+1)} - \bar{u}^{(n)}$, $\nabla^2 v = v^{(n+1)} - \bar{v}^{(n)}$. \bar{u}, \bar{v} menyatakan rata-rata nilai *optical flow* dari u, v , yang dapat diturunkan dari template *smoothing* lokal dari citra. Ukuran dari template tergantung pada kebutuhan dari citra. Berdasarkan hanya pada penghitungan, ukuran template adalah 3×3 , hasil pada persamaan (2.14) dapat ditulis sebagai berikut:

$$\begin{aligned} (I_x^2 + \lambda)u^{(n+1)} + I_x I_y v^{(n+1)} &= \lambda \bar{u}^{(n)} - I_x I_t \\ (I_y^2 + \lambda)v^{(n+1)} + I_x I_y u^{(n+1)} &= \lambda \bar{v}^{(n)} - I_y I_t \end{aligned} \quad (2.15)$$

Oleh karena itu, suatu solusi metode iterasi natural diperoleh oleh (2.15).

$$\begin{aligned} u^{(n+1)} &= \bar{u}^{(n)} - \frac{I_x(I_x \bar{u}^{(n)} + I_y \bar{v}^{(n)} + I_t)}{\lambda + I_x^2 + I_y^2} \\ v^{(n+1)} &= \bar{v}^{(n)} - \frac{I_y(I_x \bar{u}^{(n)} + I_y \bar{v}^{(n)} + I_t)}{\lambda + I_x^2 + I_y^2} \end{aligned}$$

dimana n menyatakan jumlah iterasi, u^0 dan v^0 menyatakan nilai awal dari *optical flow*, yang mengambil nol. Ketika hasil dari jarak antara dua iterasi kurang dari *threshold*, iterasi dihentikan. *Optical flow* adalah $(u^{(n+1)}, v^{(n+1)})$. Berdasarkan pada nilai dari $u^{(n+1)}$ dan $v^{(n+1)}$, kecepatan pada pixel P dinyatakan sebagai (2.16), dan juga arah pixel P nya dinyatakan sebagai (2.17).

$$f_p = \sqrt{u^{(n+1)^2} + v^{(n+1)^2}} \quad (2.16)$$

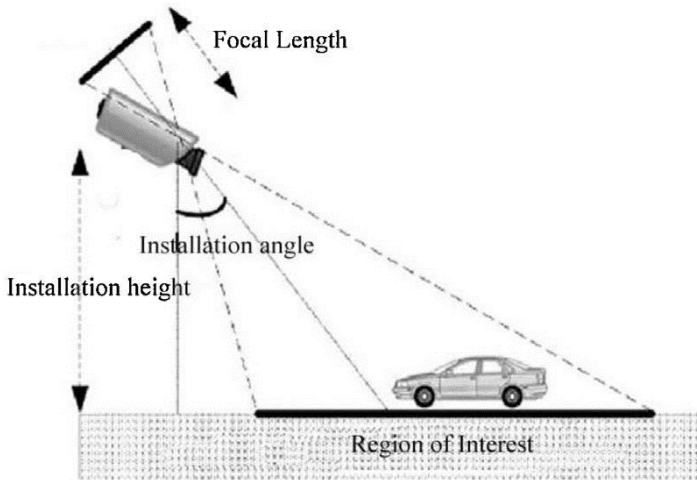
$$\theta = \arctan \frac{v^{(n+1)}}{u^{(n+1)}} \quad (2.17)$$

dengan f_p adalah nilai *optical flow*.

2.4 Pengukuran kecepatan kendaraan

Dalam sistem pengukuran kecepatan kendaraan (*vehicle speed measurement (VSM) system*), kamera merupakan mayoritas perangkat pengawasan yang digunakan. Berdasarkan pada apakah kamera perlu dikalibrasi oleh pengguna, berdasarkan pengawasan sistem VSM dapat diklasifikasikan ke dalam dua model utama: (1) model kamera yang dikalibrasi dan (2) model kamera yang tidak.

Pada model kamera yang dikalibrasi, posisi yang tepat dari kendaraan diperkirakan oleh parameter geometris dari panjang fokus, sudut instalasi dan tinggi instalasi. Gambar 2.5. Menunjukkan diagram skematik instalasi.



Gambar 2.5 Diagram skema instalasi.

Dalam citra, tidak semua citra merupakan *regions of interest*. Menyesuaikan panjang fokus, sudut instalasi dan tinggi instalasi dari kamera, digambarkan *regions of interest* (ROI) dari citra. Seperti yang ditunjukkan pada Gambar 2.6, area yang tertutup oleh garis merah adalah area yang dideteksi. Pada arah vertikal, titik perpotongan antara dua garis putih dari dari jalur ganda dan tepi dari citra adalah titik tengah dari ROI, dan *region* dapat berisi seluruh kontur mobil.

Ketika efek perspektif tidak dianggap, lebar jalan pada ROI sama dengan lebar citra (arah horizontal). Misal gambar diambil dari video 640×480 pixel, dan 7.5 meter adalah standar lebar jalan jalur ganda nasional. Sehingga rasio dari pixel pada citra dan lebar jalan dinyatakan seperti di bawah [10]:

$$k = \frac{7.5\text{m}}{640 \text{ pixel}} = \frac{7.5}{640} (\text{m/pixel})$$



Gambar 2.6 *Region of interest.*

Dalam ROI, semua target bergerak yang dideteksi ditandai oleh metode *three-frame difference* yang ditingkatkan, dan kecepatan semua titik dalam setiap blok yang bergerak dihitung, kemudian kecepatan dari target dihitung oleh kecepatan rata-rata dari semua titik dalam target. Kecepatan dari target ditunjukkan seperti (2.18).

$$f_{avr} = \frac{1}{n} \sum_{p=1}^n f_p \quad (2.18)$$

n menyatakan jumlah titik pada terget bergerak yang diperoleh oleh algoritma *three-frame difference* yang ditingkatkan, f_p menyatakan nilai geometri *optical flow* dari titik target. f_{avr} menyatakan nilai geometri *optical flow* dari blok. Berdasarkan pada nilai geometri f_p dari blok yang bergerak dan rasio k antara pixel citra dan lebar dari jalan, kecepatan kendaraan diberikan sebagai berikut:

$$V = f_{avr} \times k \quad (2.19)$$

BAB III METODE PENELITIAN

Dalam bab ini diuraikan langkah-langkah sistematis yang dilakukan dalam proses pengerjaan Tugas Akhir. Kegiatan penelitian dalam Tugas Akhir ini terdiri atas: studi literatur, analisis dan desain system, implementasi program, evaluasi dan desain program, penarikan kesimpulan dan saran, dan yang terakhir penulisan laporan tugas akhir.

3.1 Studi Literatur

Pada tahapan ini akan dilakukan pengajian tentang pengolahan video digital, pengolahan citra, *three-frame difference*, *optical flow* dengan *gray constraint* dan dideteksi kecepatan kendaraan. Studi ini dilakukan dengan membaca buku, jurnal, ataupun artikel yang terkait serta melakukan pengambilan video di jalan. Selain itu juga dilakukan dengan diskusi dengan dosen dan mahasiswa terkait.

Video tersebut diambil dengan menggunakan kamera digital. Video tersebut diambil dari atas sebuah jembatan penyeberangan yang ada di Surabaya. Rekaman video diambil berdasarkan ketentuan berikut:

- a. Jangkauan pandangan seluruh jalan arus satu arah.
- b. Menggunakan kamera digital dengan resolusi 480x640 dan 720x1280.

3.2 Analisis dan Desain Sistem

Pada tahap kedua ini akan dilakukan analisis video dan penentuan parameter-parameter apa yang akan dibutuhkan dalam pembuatan program. Kemudian dibuat desain sistem dari program sesuai dengan hasil analisa.

Berikut beberapa tahapan untuk mengolah rekaman video *off-line* arus kendaraan:

- a. Sistem ini memiliki input berupa video digital. Pemilihan area ROI (*Region of Interest*) pada rekaman video yang digunakan sebagai area pendeteksian.
- b. *Improved Three Frame Diference* digunakan untuk segmentasi setiap *frame* menjadi citra *background* dan *foreground*.
- c. Pada daerah ROI setiap objek bergerak yang masuk akan diubah menjadi *Improved Three Frame Difference*.
- d. Kemudian dilanjutkan dengan mencari nilai *Optical Flow* dari objeknya kemudian hasil yang didapat tersebut digunakan untuk mendeteksi kecepatan kendaraan.

Untuk mengimplementasikan beberapa kebutuhan yang telah dicatat pada tahap analisis. Perangkat lunak pemrograman yang dipakai untuk membuat program pendeteksian kecepatan kendaraan bergerak adalah MATLAB.

3.3 Implementasi Program

Pada tahap ketiga akan dilanjutkan dengan implementasi sistem dalam bentuk perangkat lunak sesuai dengan hasil analisis dan desain sistem. Akan dibuat desain *interface* yang menarik dan *user friendly* untuk memudahkan dan membuat nyaman pengguna.

3.4 Uji Coba dan Evaluasi Program

Pada tahap ini dilakukan pengujian terhadap perangkat lunak yang telah selesai dibuat menggunakan *input* video rekaman kendaraan bergerak di jalan. Kemudian dilakukan analisis pembahasan sehingga dapat dicatat beberapa hal yang dijadikan pertimbangan dalam menarik kesimpulan. Pada tahap ini juga dilakukan evaluasi terhadap program yang telah dibangun. Hasil evaluasi dicatat untuk membenahi hal-hal yang masih kurang. Kemudian dilanjutkan dengan uji coba deteksi kecepatan kendaraan, data hasil deteksi kecepatan dari program akan dibandingkan dengan kecepatan data sebenarnya. Kemudian dihitung tingkat akurasi dari program menggunakan

persamaan (3.1). Selain itu akan dilakukan evaluasi sehingga dapat membenahi hal-hal yang masih kurang dan mengatasi error yang ditemukan.

$$PK = \frac{KS - |(KS - KP)|}{KS} \times 100\% \quad (3.1)$$

Dengan PK adalah prosentase keberhasilan, KP adalah kecepatan kendaraan yang terdeteksi dengan benar, dan KS adalah kecepatan kendaraan sebenarnya.

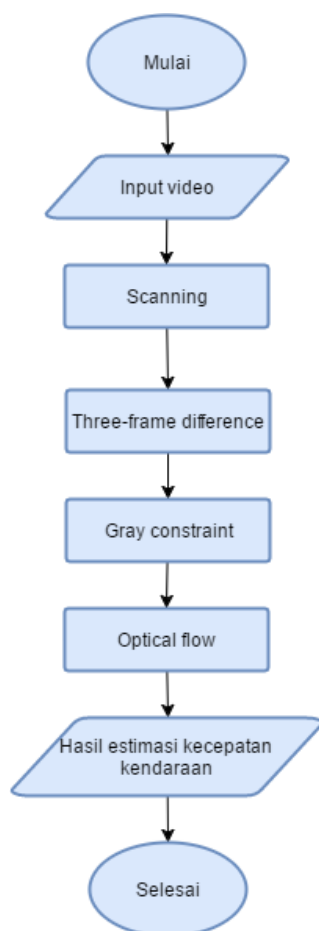
3.5 Penarikan Kesimpulan dan Saran

Untuk tahap kelima ini akan ditarik kesimpulan dari hasil pengerjaan, uji coba, dan pembahasan serta disampaikan saran untuk pengembangan selanjutnya.

3.6 Penulisan Laporan Tugas Akhir

Pada tahap keenam ini merupakan tahap yang terakhir. Penulis akan menuliskan semua hasil yang telah didapatkan selama pengerjaan tugas akhir ini.

Blok diagram dari proses pengukuran kecepatan kendaraan yang ditunjukkan oleh gambar berikut.



Gambar 3.1 Blok diagram sistem pengukuran kecepatan kendaraan.

BAB IV

PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini akan dibahas mengenai perancangan dan implementasi sistem dimulai dari pembahasan proses pengambilan data masukan, pengolahan data masukan dengan algoritma *Optical Flow* Horn-Schunck dengan metode *Improved Three Frame Difference*, serta penjelasan mengenai cara untuk mendapatkan data keluaran yang sesuai dengan tujuan dari penelitian Tugas Akhir ini.

Selain hal yang telah disebutkan di atas, perancangan sistem juga meliputi perancangan antar muka (*user interface*) untuk memudahkan penelitian dalam memproses data dan melakukan analisis terhadap keluaran dari sistem yang telah dibangun.

Hasil dari analisis dan perancangan sistem dilanjutkan dengan implementasi sistem. Sehingga perangkat lunak yang dibutuhkan untuk memproses informasi dari rekaman video digital dapat dilakukan.

4.1 Analisis Sistem

Untuk membangun perangkat lunak yang dapat mendeteksi kecepatan kendaraan bergerak berbasis video digital. Maka diperlukan sistem yang dapat mengolah data masukan berupa rekaman kendaraan di jalan raya, pada kasus Tugas Akhir ini menggunakan data masukan berupa video digital arus kendaraan secara *offline*. Dengan mengkombinasikan algoritma *Optical Flow* Horn-Schunck dengan metode *Improved Three Frame Difference* [10] untuk mendapatkan objek kendaraan dalam video yang nantinya akan digunakan dalam mendeteksi kecepatan kendaraan maka akan tercipta sistem yang sesuai dengan tujuan dari penelitian Tugas Akhir ini.

4.1.1 Analisa Sistem Perangkat Lunak

Perangkat lunak yang akan dibangun dapat digunakan oleh dinas perhubungan untuk mengamati kondisi arus lalu lintas di

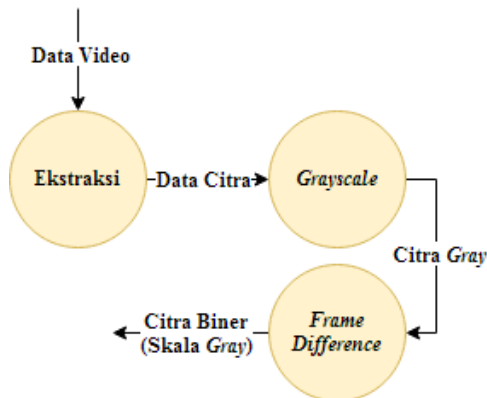
suatu jalan sehingga dapat menganalisa kebutuhan infrastruktur jalan tersebut. Selain itu masyarakat juga dapat memperoleh informasi kondisi lalu lintas pada suatu jalan. Adapaun *Data Flow Diagram* dari aplikasi tersebut adalah sebagai berikut:



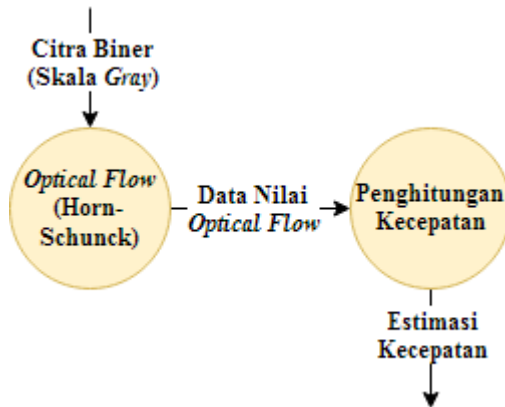
Gambar 4.1 DFD level 0 program estimasi kecepatan kendaraan.



Gambar 4.2 DFD level 1 program estimasi kecepatan kendaraan.



Gambar 4.3 DFD Level 2 dari proses segmentasi.



Gambar 4.4 DFD Level 2 dari proses penghitungan kecepatan.

Sistem perangkat lunak yang dibangun ini memiliki beberapa tahapan sebagai berikut:

a. Akuisisi video

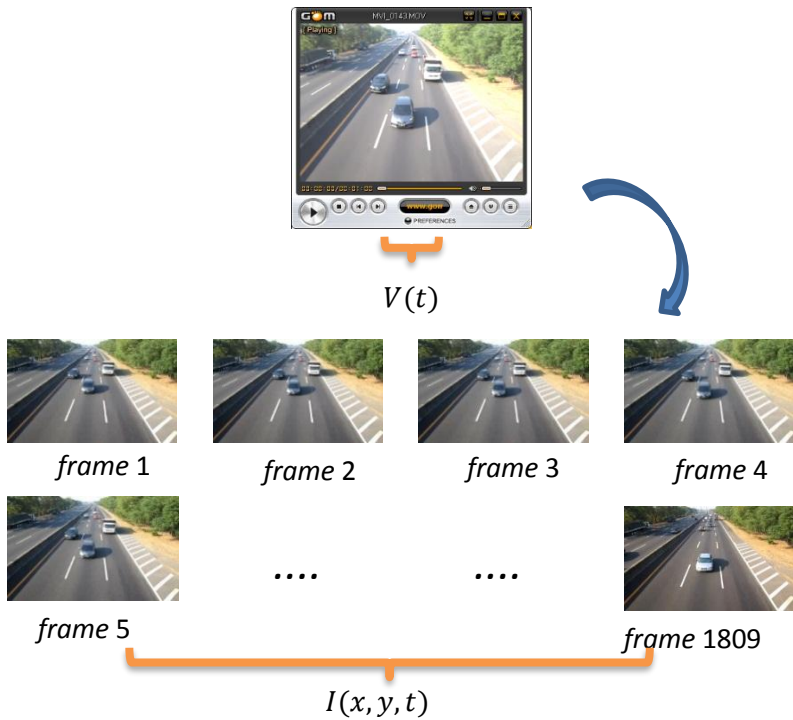
Data masukan berupa rekaman video digital *offline* arus kendaraan yang kemudian di-*scanning*. *Scanning* adalah proses pemecahan video menjadi beberapa rangkaian citra yang sering disebut dengan *frame*.

Pengambilan video arus lalu lintas kendaraan di Jalan Kedung Cowek dan Jembatan Surabaya, Surabaya. Video diambil menggunakan kamera DSLR Canon. Video yang diambil berdasarkan ketentuan berikut:

- Sudut pandang yang diambil ketika merekam adalah dari atas dengan jangkauan pandangan seluruh jalan satu arah.

b. Penentuan ROI (*Region of Interest*)

ROI (*Region of Interest*) akan digunakan sebagai area pendeteksian kecepatan kendaraan bergerak.



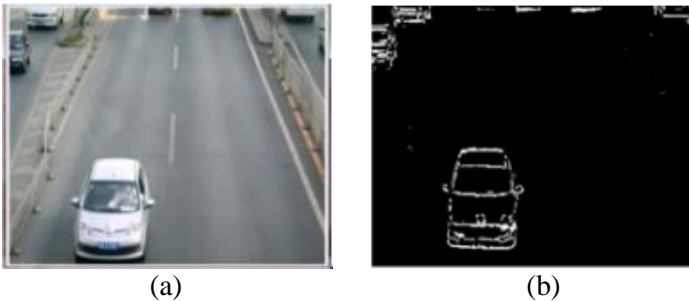
Gambar 4.5 Proses *scanning* video.



Gambar 4.6 Citra dengan ROI (daerah didalam kotak merah).

c. Segmentasi citra

Kemudian sistem membagi *frame* menjadi citra *background* dan citra *foreground* dengan menggunakan algoritma *Improved Three Frame Difference* [10]. Citra *background* adalah citra yang menjadi latar belakang dari suatu objek yang bergerak. Sedangkan *foreground* adalah gambaran objek bergerak yang terdeteksi.



Gambar 4.7 Citra hasil *Background Subtraction* dengan metode *Improved Three Frame Difference*. (a) citra asli, (b) citra *foreground*.

d. Proses deteksi kecepatan kendaraan

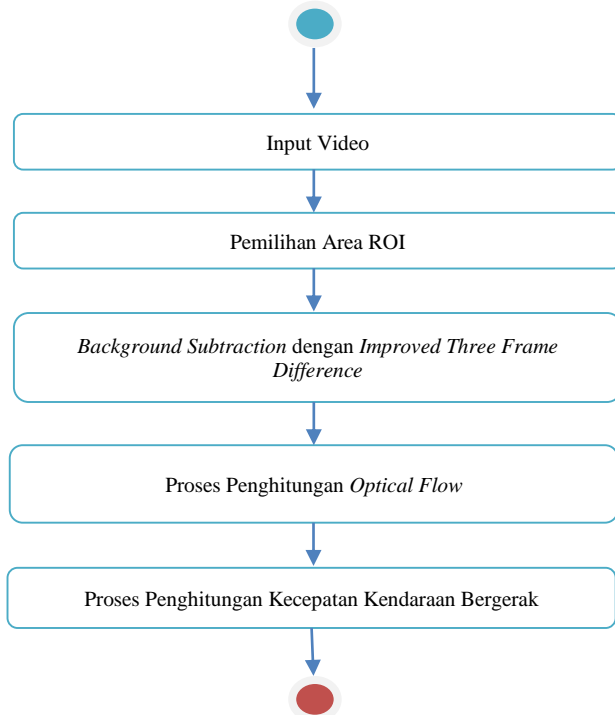
Objek hasil algoritma *Improved Three Frame Difference* tersebut akan dideteksi kecepatannya menggunakan *Optical Flow*. Nilai *Optical Flow* dari objek tersebut dihitung, sehingga didapatkan kecepatannya.

4.1.2 Analisis Kebutuhan Sistem

Perangkat lunak ini dibangun menggunakan bahasa pemrograman MATLAB 2016b dan 2017a.

Selain itu juga diperlukan kamera digital untuk merekam arus kendaraan di jalan. Program ini dikembangkan menggunakan komputer dengan spesifikasi Intel Core i5-2400S CPU @2.50 GHz 4.00GB RAM.

Gambar 4.8 di bawah ini menjelaskan tentang gambaran pokok *business rule* dari perangkat lunak yang dibangun.



Gambar 4.8 *Business Rule* sistem perangkat lunak.

4.2 Perancangan Sistem

Perancangan sistem meliputi perancangan data sistem, perancangan proses, dan perancangan antar muka sistem. Rincian dari proses perancangan sistem tersebut dapat dilihat di bawah.

4.2.1 Perancangan Data Sistem

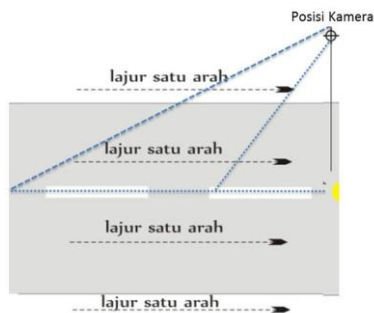
Terdapat tiga macam data yang digunakan oleh sistem ini antara lain data masukan, data proses, dan data keluaran. Pada Tugas Akhir ini data masukan berupa video rekaman *offline* di

beberapa jalan diambil secara mandiri oleh peneliti. Data proses merupakan data yang berisi parameter-parameter yang akan digunakan oleh algoritma *Improved Three Frame Difference* serta metode *speed detection*. Sedangkan data keluaran adalah informasi mengenai kecepatan kendaraan bergerak yang terdeteksi oleh sistem.

A. Data Masukan

Data masukan sistem ini berupa video *offline* rekaman kendaraan di jalan yang diambil menggunakan kamera digital. Video diambil dengan posisi lensa kamera menjangkau ke seluruh area jalan satu arah. Dalam perekaman video tersebut, data diambil dari atas sebuah jembatan penyeberangan orang. *Layout* untuk pengambilan video rekaman dapat dilihat pada Gambar 4.9. File video yang diambil memiliki spesifikasi sesuai dengan batasan masalah yang terdapat dalam penelitian Tugas Akhir ini yaitu:

- a. Format ekstensi rekaman video *offline* dapat berupa: *.mp4.



Gambar 4.9 *Layout* untuk pengambilan video.

B. Data Proses

Data proses merupakan data yang digunakan dalam proses pengolahan data masukan. Data proses ini diperoleh dari hasil pengolahan data masukan sesuai dengan tahapan algoritma dan

metode yang telah disusun. Tabel 4.1 menjelaskan tahapan dari data proses.

Tabel 4.1 Tabel data proses.

No	Tahapan	Input	Output
1.	<i>Input Awal</i>	Video	<i>Frame Citra</i>
2.	Pilih Area ROI	<i>Frame Citra</i>	Citra ROI
3.	Hitung skala lebar jalan dengan lebar citra	Lebar jalan	Skala perbandingan lebar jalan dan lebar citra
4.	Proses <i>Improved Three Frame Difference</i>	Citra ROI	Citra <i>Foreground</i>
5.	Hitung <i>Optical Flow</i>	Objek hasil proses <i>Improved Three Frame Difference</i>	Jumlah nilai <i>Optical Flow</i> dari objek gerak
6.	Hitung Kecepatan Kendaraan	Jumlah nilai <i>Optical Flow</i> dari objek gerak dan skala perbandingan lebar jalan dan lebar citra	Kecepatan kendaraan

C. Data Keluaran

Data keluaran berupa hasil deteksi kecepatan kendaraan bergerak. Program ini juga menghasilkan keluaran berupa citra ROI dan citra *foreground*.

4.2.2 Perancangan Proses

A. Perancangan Proses Algoritma

Proses algoritma ini dimulai dengan pemilihan ROI (*Region of Interest*) yang digunakan sebagai area pendeteksian

kecepatan kendaraan bergerak. Selanjutnya citra ROI tersebut diproses menggunakan algoritma *Background Subtraction* dengan metode *Improved Three Frame Difference* untuk mengelompokkan bagian piksel yang tergolong citra *foreground* atau *background* [10]. Citra *foreground* yang telah didapat akan dihitung nilai *Optical Flow*-nya menggunakan algoritma *Optical Flow* Horn-Schunck. Pada tahap akhir objek yang terdeteksi dideteksi kecepatannya.

Gambar 4.10 menjelaskan diagram alir sistem perangkat lunak.

B. Perancangan Proses *Improved Three Frame Difference*

Berikut adalah alur proses untuk mengolah citra per *frame* menggunakan metode *Improved Three Frame Difference*. Berikut adalah blok diagram proses *Improved Three Frame Difference*.

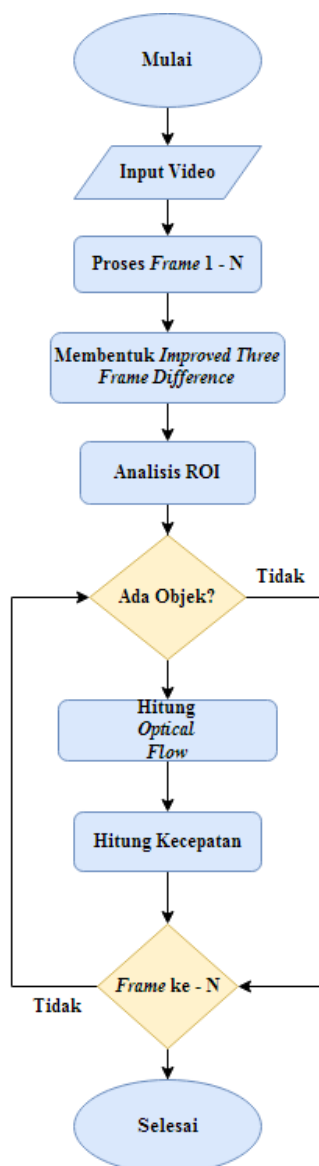
C. Perancangan Proses Deteksi Kecepatan Kendaraan

i. Pengukuran Skala (Meter/ Pixel)

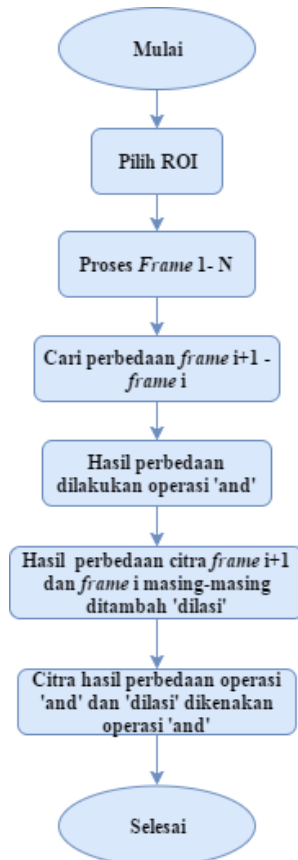
Untuk mengukur kecepatan sebuah obyek pada tugas akhir ini, harus diketahui informasi atas lebar jalan yang digunakan dalam uji coba dan lebar citra. Lebar jalan raya umumnya adalah 7,5 meter. Kemudian lebar jalan yang didapat dibagi dengan lebar citra.

ii. Menghitung Nilai *Optical Flow*

Untuk mendapatkan nilai *optical flow*, citra yang digunakan harus dalam *gray scale*. Citra hasil dari metode *improved three frame difference* sudah dalam bentuk *gray scale* sehingga nilai *optical flow* bisa langsung didapat dengan menggunakan metode perhitungan *optical flow* Horn-Schunck.



Gambar 4.10 Diagram alir sistem perangkat lunak.



Gambar 4.11 Diagram alir *Improved Three Frame Difference*.

iii. Deteksi Kecepatan

Deteksi kecepatan kendaraan bergerak merupakan langkah akhir dari semua proses, dengan tujuan mendapatkan informasi berapa kecepatan kendaraan tersebut dalam satuan km/jam. Setelah didapatkan skala (meter/pixel) dan nilai rata-rata *optical flow* dari metode *optical flow* Horn-Schunck maka bisa dideteksi kecepatannya menggunakan persamaan (2.19).

Proses deteksi kecepatan kendaraan dengan cara mengolah objek-objek hasil *Improved Three Frame Difference* pada ROI. Untuk mengukur kecepatan sebuah obyek, harus diketahui informasi atas variabel yang berpengaruh, diantaranya skala dan nilai *optical flow*. Proses mencari skala dilakukan dengan mengukur lebar jalan yang digunakan dalam uji coba. Biasanya, lebar sebuah jalan ganda standar adalah 7,5 meter. Skala didapat dari lebar jalan dibagi dengan lebar citra.

Input yang dibutuhkan untuk proses deteksi kecepatan adalah:

- a. Lebar jalan
- b. Lebar citra

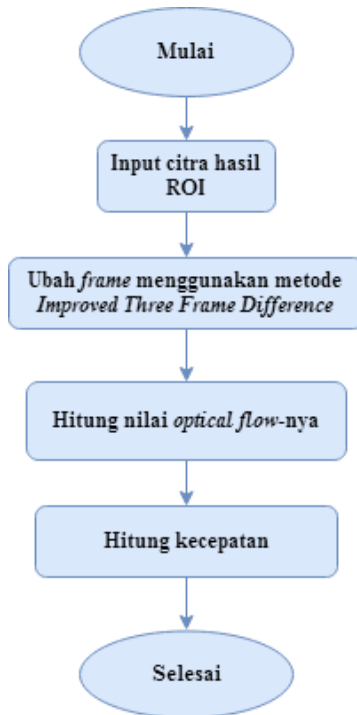
Diagram alir pendeteksian kecepatan kendaraan dapat dilihat pada Gambar 4.12.

4.2.3 Perancangan Antar Muka Sistem

Desain antar muka sistem dibutuhkan agar pengguna dengan mudah mengoperasikan perangkat lunak yang dibangun. Desain ini dibuat semenarik mungkin dan *user friendly*. Berikut adalah desain antar muka halaman utama dan halaman detail proses:

A. Perancangan Halaman Utama

Halaman utama ini memiliki desain antar muka yang ditunjukkan pada Gambar 4.12. Pada halaman ini dilakukan proses pemilihan video, pemilihan ROI, dan mengisi parameter proses. Halaman ini menampilkan video hasil deteksi kecepatan kendaraan. Antar muka halaman utama terdiri dari:



Gambar 4.12 Diagram alir deteksi kecepatan.

1. *Axis1*, berfungsi untuk menampilkan frame dari video input.
2. *Axis2*, berfungsi untuk menampilkan objek yang akan dilacak pergerakannya. Objek ini diperoleh setelah pengguna memilih ROI objek pada frame pertama yang ditampilkan pada *axis1*.
3. *Push button Browse*, berfungsi untuk memilih video input yang akan digunakan dalam proses pelacakan.
4. *Push button Start*, berfungsi untuk menjalankan video dan juga untuk melakukan proses pendeteksian kecepatan.
5. *Push button Reset* digunakan untuk menghapus semua historis dari proses sebelumnya.

6. *Push button Hasil* digunakan untuk menghitung beberapa parameter output yakni lebar jalan.
7. *Edit text Lebar Jalan*, berfungsi agar pengguna dapat menentukan ukuran jalan yang digunakan (umumnya 7,5 meter), sehingga didapat skala.
8. *Push button Close*, digunakan untuk menutup program.
9. *Text*, berfungsi untuk menampilkan kecepatan objek yang diamati.



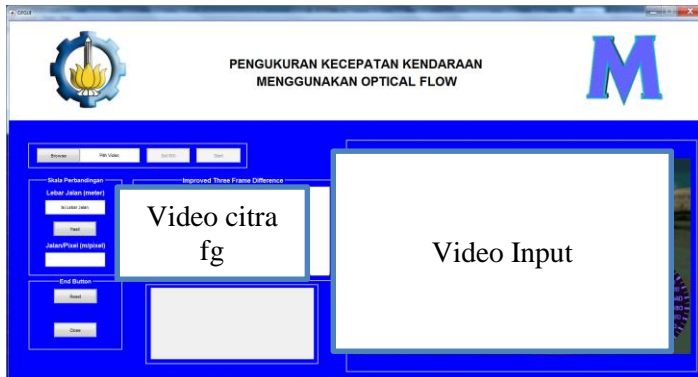
Gambar 4.13 Antar muka halaman utama.

B. Perancangan Halaman Detail

Pada halaman detail ini ditampilkan video *input*, video citra *foreground*. Desain antar muka dari halaman detail ditunjukkan pada Gambar 4.14.

4.3 Implementasi Sistem

Implementasi sistem meliputi implementasi *input* video, implementasi pemilihan area ROI, implementasi proses *improved three frame difference*, implementasi pendeteksian kecepatan kendaraan. Rincian dari implementasi tersebut dijelaskan di bawah.



Gambar 4.14 Antar muka halaman detail.

4.3.1 Implementasi *Input* Video

Pada proses ini akan ditentukan video yang nantinya digunakan sebagai data masukan untuk diproses. Penjabaran tentang proses pemilihan video adalah sebagai berikut:

- Fungsi : menginputkan video bertipe *.avi, *.mp4, dll. pada form simulasi
- Input : video bertipe *.avi, *.mp4, dll.
- Deskripsi : mengambil video bertipe *.avi, *.mp4, dll. yang telah disimpan pada komputer.

Gambar 4.14 adalah tampilan antar muka pengambilan *input* video. Kode program untuk menerima *input* video adalah sebagai berikut:

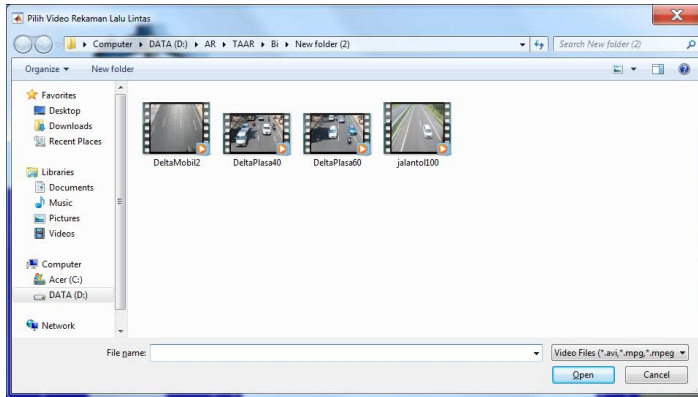
```
[filename, pathname] = uigetfile( ...
{'*.avi;*.mpg;*.mpeg;*.mp4', 'Video Files
(*.avi,*.mpg,*.mpeg,*.mp4)';
'*.*', 'All Files (*.*)'}, ...
'Pilih Video Rekaman Lalu Lintas');
if ~isequal(filename,0)
    fullname = fullfile(pathname, filename);
    vid = fullname;
```

```

else
    return
end

```

Kode program lebih lengkap dapat dilihat pada lampiran.



Gambar 4.15 Antar muka *input* video.

4.3.2 Implementasi Pemilihan area ROI

Region of Interest (ROI) adalah daerah bagian dari citra atau *frame* yang akan diproses pada sistem ini. Pemilihan ROI ini bertujuan untuk memudahkan dan membantu program dalam menspesifikasikan kebutuhan program. Ukuran ROI pada sistem ini dianjurkan memuat setengah dari ukuran *frame* inputan. Sehingga dapat menangkap objek kendaraan secara utuh dan hasil yang diinginkan lebih akurat dan lebih efisien. Hal ini dikarenakan program hanya mengolah informasi dari bagian piksel yang dibutuhkan saja yaitu ROI. Kode program untuk pemilihan ROI adalah sebagai berikut:

```

global position
h = imrect;
setColor(h, 'red');
position = wait(h);

```

```

if ~isempty(position)
    set(handles.pbStart,'Enable','on');
end

```

Kode program selengkapnya disajikan pada Lampiran. Gambar 4.16 merupakan tampilan program setelah pemilihan ROI dilakukan.



Gambar 4.16 Kotak merah pada *frame* tersebut adalah area ROI yang dipilih.

4.3.3 Implementasi Proses *Improved Three Frame Difference*

Proses *Improved Three Frame Difference* digunakan untuk mencari citra *foreground* dan *background*. Citra *foreground* merupakan citra yang merepresentasikan keadaan objek yang bergerak. Kode program *Improved Three Frame Difference* dalam Tugas Akhir ini dapat dilihat pada Lampiran.

```

function im_tri_diff_frame =
im_tri_frame_diff(tri_diff_frame,
dil_tri_diff_frame)

```

```

global HEIGHT LENGTH

```

```

for i = 1 : HEIGHT

```

```

    for ii = 1 : LENGTH
        if (tri_diff_frame(i, ii) ~=
dil_tri_diff_frame(i, ii))
            im_tri_diff_frame(i, ii) = 255;
        else
            im_tri_diff_frame(i, ii) = 0;
        end
    end
end
end

```

4.3.4 Implementasi Pendeteksian Kecepatan Kendaraan

Implementasi ini merupakan tahap terakhir dan juga merupakan fungsi utama program.

Objek-objek hasil dari analisis ROI akan dideteksi kecepatannya. Kode program untuk mengetahui nilai *optical flow* kendaraan yang melewati daerah ROI adalah sebagai berikut:

```

global LENGTH k nz

opticFlow = opticalFlowHS('MaxIteration',
inf, 'VelocityDifference', 1);
flow = estimateFlow(opticFlow, citfd);
nz = nnz(flow.Magnitude);
fpm = sum(sum(flow.Magnitude));
v = fpm*k*3.6;

```

Kode selengkapnya pada Lampiran.

Berikut ini kode program pendeteksian kecepatan kendaraan selama melewati daerah ROI:

```

global reader position nz
kec = 0;
n = 0;
while ~isDone(reader)
    citfd = framedif();

```



```

axes(handles.axesFramedif);
imshow(citfd);
drawnow

v = opticalflow(citfd);

if nz > 500
    kec = kec + v;
    n = n+1;
end

axes(handles.axesFrame);
imshow(step(reader));
drawnow
end

kec = kec/n;
set(handles.textSpeed, 'String', kec);

```

Kode selengkapnya terdapat pada Lampiran. Setelah perhitungan selesai dilakukan, maka nilainya akan ditampilkan di dalam GUI. Gambar 4.17 adalah antar muka dari hasil deteksi kecepatan kendaraan.



Gambar 4.17 Antar muka aplikasi saat pendeteksian kecepatan kendaraan.

BAB V

UJI COBA DAN PEMBAHASAN

Bab ini menjelaskan mengenai pengujian program dan pembahasan dari hasil uji coba. Pengujian yang dilakukan disini adalah pengujian program dengan *input* video rekaman kendaraan bergerak di jalan raya tepatnya di Jalan Kedung Cowek dan Jembatan Surabaya, Surabaya. Pengambilan video dilakukan pada waktu pagi dan sore hari. Durasi rekaman yang digunakan pada pengujian ini adalah 1 sampai 6 detik dengan objek yang direkam dalam video tersebut adalah pengendara sepeda motor.

5.1 Data Uji Coba

Uji coba pada program dalam Tugas Akhir ini dilakukan terhadap video *.mp4 dan *.avi. Video uji coba sudah tersimpan dalam penyimpanan komputer dan diperoleh dari hasil rekaman. Video yang digunakan sebanyak 11 video. Daftar input uji coba tersebut antara lain disajikan dalam Tabel 5.1.

Tabel 5.1 Tabel Data Video yang Digunakan.

No	Nama	<i>Framerate</i>	Pixel
1.	SM20a.mp4	30	720x1280
2.	SM20b.mp4	30	720x1280
3.	SM20c.mp4	30	720x1280
4.	SM30b.mp4	30	720x1280
5.	JS40a.mp4	30	720x1280
6.	JS40b.mp4	30	720x1280
7.	JS40c.mp4	30	720x1280
8.	JS50a.mp4	30	720x1280
9.	JS50b.mp4	30	720x1280
10.	dtc-30-50d.avi	30	480x640
11.	dtc-40-50d.avi	30	480x640

5.2 Uji Coba Pendeteksian Kecepatan Kendaraan Bergerak

Pengujian program ini dilakukan dengan *input* video rekaman kendaraan bergerak di jalan raya Kedung Cowek, Jembatan Surabaya, dan Wonokromo, Surabaya. Pengambilan video dilakukan pada waktu pagi dan sore hari. Durasi rekaman yang digunakan adalah 1-6 detik dengan objek yang direkam dalam video tersebut adalah sepeda motor. Kendaraan yang melewati kamera berjalan dengan kecepatan yang relatif konstan.

Pengujian dilakukan dengan memasukkan beberapa nilai parameter, diantaranya yaitu:

1. Lebar Jalan = 7.5 meter
2. Lebar Citra

Setelah sebuah parameter itu dimasukkan selanjutnya akan dilakukan proses perhitungan untuk mendapatkan hasil skala perbandingan jalan dengan lebar citra pada video. Kemudian dilakukan perhitungan kecepatannya yaitu dengan menjalankan hasil rekaman video pada program. Saat kendaraan berjalan dan kemudian memasuki daerah ROI maka pada saat kendaraan berada pada daerah ROI tersebut akan dilakukan proses perhitungan untuk mendapatkan kecepatan dari kendaraan tersebut. Proses perhitungan disini didapat dari proses penghitungan nilai *optical flow* dari kendaraan yang kemudian dikali dengan paramater yang sudah didapatkan pada proses sebelumnya. Maka akan diperoleh kecepatan kendaraannya yang ditampilkan pada GUI-nya.

5.2.1 Rekaman Video dengan Kecepatan 20 Km/Jam

Data ini adalah data yang didapatkan oleh penulis dengan melakukan perekaman pada jalan raya, tepatnya di jalan Kedung Cowek, Surabaya. Data tersebut diambil dari atas sebuah jembatan penyeberangan orang di daerah tersebut menggunakan kamera digital pada saat pagi hari.

Tabel 5.2 Tabel Pendeteksian Kecepatan Kendaraan dengan Kecepatan 20 Km/Jam.

Nama Video	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)
SM20a	30	20	22,66
SM20b	30	20	22,84
SM20c	30	20	21,42

5.2.2 Rekaman Video dengan Kecepatan 30 Km/Jam

Data ini adalah data yang diambil ditempat yang sama dan waktu yang sama yakni pagi hari seperti data pada sub-bab 5.2.1. Namun kecepatan relatif kendaraan yang diuji untuk diestimasi oleh program 30 Km/Jam.

Tabel 5.3 Tabel Pendeteksian Kecepatan Kendaraan dengan Kecepatan 30 Km/Jam.

Nama Video	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)
SM30b	30	30	33,31

5.2.3 Rekaman Video dengan Kecepatan 40 Km/Jam

Data ini adalah data yang didapatkan dengan melakukan perekaman pada jalan raya, tepatnya di jalan Jembatan Surabaya, Kenjeran Baru, Surabaya. Data tersebut diambil dari atas sebuah anjungan yang terdapat pada jalan tersebut. Adapun perekaman dilakukan menggunakan kamera digital dan pada saat sore hari.

Tabel 5.4 Tabel Pendeteksian Kecepatan Kendaraan dengan Kecepatan 40 Km/Jam.

Nama Video	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)
JS40a	30	40	35,23
JS40b	30	40	34,50
JS40c	30	40	35,75

5.2.4 Rekaman Video dengan Kecepatan 50 Km/Jam

Data rekaman video dengan kecepatan 50 Km/Jam didapatkan di tempat dan waktu yang sama yakni sore seperti pada data yang disebutkan pada sub-bab 5.2.3.

Tabel 5.5 Tabel Pendeteksian Kecepatan Kendaraan dengan Kecepatan 50 Km/Jam.

Nama Video	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)
JS50a	30	50	44,26
JS50b	30	50	37,42

5.2.5 Rekaman Video dengan Kecepatan 30 dan 40 Km/Jam

Data ini adalah data lain yang penulis dapat dari Danang Wahyu Wicaksono, S.Si., M.Si. pada penelitiannya saat menempuh pendidikan S2 Matematika di Institut Teknologi Sepuluh Nopember (ITS). Data tersebut diambil dari atas sebuah jembatan penyeberangan orang di jalan Wonokromo, Surabaya. Adapun rincian data tersebut sudah dijelaskan sebelumnya pada sub-bab 5.1. Namun ada sedikit tambahan mengenai rinciannya yakni data tersebut diambil dengan sudut pengambilan sebesar 50 derajat [15].

Tabel 5.6 Tabel Pendeteksian Kecepatan Kendaraan dengan Kecepatan 30 dan 40 Km/Jam.

Nama Video	Framerate	Sebenarnya (km/jam)	Program (km/jam)
dtc-30-50d.avi	30	30	26,33
dtc-40-50d.avi	30	40	34,86

5.2.6 Pembahasan Uji Coba Pendeteksian Kecepatan Kendaraan

Berdasarkan hasil uji coba yang sudah dilakukan, program dapat mengukur kecepatan kendaraan bergerak dengan baik pada video SM20a.mp4, SM20b.mp4, SM20c.mp4, SM30b.mp4, JS40a.mp4, JS40b.mp4, JS40c.mp4, JS50a.mp4, JS50b.mp4, dtc-30-50d.avi, dan dtc-40-50d.avi. Hal itu terlihat dari Tabel 5.2, 5.3, 5.4, 5.5, dan 5.6, dimana penghitungan program hampir mendekati dengan kecepatan kendaraan sebenarnya.

Dari data yang tersaji pada Tabel 5.2, 5.3, 5.4, 5.5, dan 5.6, dapat dihitung tingkat akurasi keberhasilan program dalam mendeteksi kecepatan kendaraan bergerak. Penghitungan tingkat akurasi keberhasilan ini didasarkan pada persamaan (3.1):

$$PK = \frac{KS - |KS - KP|}{KS} \times 100\%$$

dengan,

PK = Prosentase keberhasilan,

KS = Kecepatan kendaraan sebenarnya,

KP = Kecepatan kendaraan yang terdeteksi program.

Maka, tingkat akurasi keberhasilan program dalam mengukur kecepatan kendaraan bergerak dapat dilihat pada Tabel 5.7.

Tabel 5.7 Tabel Prosentase Keberhasilan Video Uji Coba

Nama Video	Sebenarnya (km/jam)	Program (km/jam)	PK	<i>Run time</i> (detik)
SM20a.mp4	20	22,66	86,7%	37,7
SM20b.mp4	20	22,84	85,8%	48,0
SM20c.mp4	20	22,42	87,9%	18,3
SM30b.mp4	30	33,31	88,97%	9,3
JS40a.mp4	40	35,23	88,07%	22,2
JS40b.mp4	40	34,50	86,25%	21,3
JS40c.mp4	40	35,75	89,37%	23,7
JS50a.mp4	50	44,26	88,52%	18,5
JS50b.mp4	50	37,42	74,84%	14,9
dtc-30-50d.avi	30	26,33	87,77%	11,4
dtc-40-50d.avi	40	34,86	87,15%	8,5

Dari data prosentase keberhasilan pada Tabel 5.7 program dapat mendeteksi kecepatan kendaraan bergerak dengan baik.

Pada Tabel tersebut juga ditunjukkan bahwa tingkat akurasi pada program paling rendah sebesar 74,84% dan tingkat akurasi paling tinggi 89,37%.

BAB VI

PENUTUP

Bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan. Di samping itu, pada bab ini juga dimasukkan beberapa saran yang dapat digunakan jika penelitian ini ingin dikembangkan.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian program, maka dapat diambil kesimpulan sebagai berikut:

1. Program dapat mendeteksi kecepatan kendaraan bergerak dari video rekaman dengan membuat *background* dan *foreground* menggunakan metode *Improved Three Frame Difference*. Tahapan dalam penghitungan ini adalah:
 - a. Pemilihan ROI (*Region of Interest*) sebagai area pendeteksian kecepatan kendaraan.
 - b. Proses algoritma *Improved Three Frame Difference* untuk mencari *background* pada setiap *frame* sehingga didapatkan citra *foreground*.
 - c. Kemudian dilakukan penghitungan menggunakan *optical flow* Horn-Schunck pada citra *foreground* yang telah di *Improved Three Frame Difference* untuk tahap pendeteksian kecepatan kendaraan.
2. *Optical flow* Horn-Schunck dapat melakukan estimasi kecepatan kendaraan dengan baik. Hal tersebut dapat dilihat dari hasil uji coba yang sudah dilakukan prosentase keberhasilan paling kecil hanya 86,70% untuk kendaraan dengan kecepatan relatif konstan 20 km/jam.

6.2 Saran

Dengan melihat hasil yang dicapai pada penelitian ini, ada beberapa hal yang penulis sarankan untuk pengembangan selanjutnya yaitu:

1. Program belum dapat mendeteksi kendaraan bergerak untuk *multiple objects*. Penelitian ke depan diharapkan dapat

- mengembangkan mendeteksi kecepatan untuk *multiple objects*.
2. Program hanya dapat mengestimasi kecepatan kendaraan. Penelitian kedepan diharapkan program mampu untuk melakukan penghitungan kendaraan, pengklasifikasian kendaraan, dan program mampu digunakan di setiap waktu (pagi, siang, sore, dan malam).

DAFTAR PUSTAKA

- [1] Celik, T., Kusetogullari, H. (2010). "Solar-Powered Automated Road Surveillance System for Speed Violation Detection". **IEEE Transactions on Industrial Electronics**, Vol. **57**, Issue **9**, Hal. 3216-3227.
- [2] Mei, T.X., Li, H. (2010). "Measurement of Absolute Vehicle Speed with a Simplified Invers Model". **IEEE Transactions on Vehicular Technology**, Vol. **59**, Issue **3**, Hal. 1164-1171.
- [3] Damien, L-D., Christophe, G. (2009). "Doppler-Based Ground Speed Sensor Fusion and Slip Control for a Wheeled Rover". **IEEE/ASME Transactions on Mechatronics**, Vol. **14**, Issue **4**, Hal. 484-492.
- [4] Fernandez-Caballero, A., Gomez, F.J., Lopez-Lopez, J. (2008). "Road-Traffic Monitoring by Knowledge-Driven Static and Dynamic Image Analysis". **Expert Systems with Applications**, Vol. **35**, Issue **3**, Hal. 701-719.
- [5] Li, Y., Yin, L., Jia, Y. (2008). "Vehicle Speed Measurement Based on Video Images". In: **2008 3rd International Conference on Innovative Computing Information and Control (ICICIC)**, Hal. 439.
- [6] Canals, R., Roussel, A., Famechon, J.-L., Treuillet, S. (2002). "A Biprocessor-Oriented Vision-Based Target Tracking System". **IEEE Transactions on Industrial Electronics**, Vol. **49**, Issue **2**, Hal. 500-506.

- [7] Cucchiara, R., Grana, C., Piccardi, M., Prati, A. (2003). "Detecting Moving Objects, Ghosts, and Shadows in Video Streams". **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 25, Issue 10, Hal. 1337-1342.
- [8] Schoepflin, T., Dailey, D. (2003). "Dynamic Camera Calibration of Roadside Traffic Management Cameras for Vehicle Speed Estimation". **IEEE Transactions on Intelligent Transportation Systems**, Vol. 4, Issue 2, Hal. 90-98.
- [9] Zhang, Z.-D., Zhang, C.-w. (2010) "Method to Improve the Accuracy of Video-Based Vehicle Speed Measurement". **Journal of Shanghai Jiaotong University**, Vol. 44, Issue 10, Hal. 1439-1443.
- [10] Lan, Jinhui, Li, Jian, Hu, Guangda, Ran, Bin, Wang, Ling. (2013). "Vehicle Speed Measurement Based on Gray Constraint Optical Flow". **International Journal for Light and Electron Optics**, Vol. 125, Issue 1, Hal. 289-295.
- [11] Weng, M., Huang, G., Da, X. (2010). "A New Interframe Difference Algorithm for Moving Target Detection". **Proceedings of International Congress on Image and Signal Processing**, Hal. 285-289.
- [12] Lookingbill, A., Rogers, J., Lieb, D., Curry, J. dkk. (2007). "Reverse Optical Flow for Self-Supervised Adaptive Autonomous Robot Navigation". **International Journal of Computer Vision**, Vol. 74, Issue 74, Hal. 287-302.

- [13] Lucas, B. D., Kanade, T. (1981). "An Iterative Image Registration Technique with an Application to Stereo Vision". **Proc. Seventh Int'l Joint Conf. Artificial Intelligence**, Hal. 674-679.
- [14] Horn, B.K.P., Schunck, B.G. (1981). "Determining Optical Flow". **Artificial Intelligence, Vol. 16, Issue 1-3**, Hal. 185-203.
- [15] Wicaksono, D. W. (2017). **Pengembangan Sistem Estimasi Kecepatan pada Kedaraan Bergerak Berbasis Pengolahan Citra Digital**. (Thesis). Surabaya: Institut Teknologi Sepuluh Nopember.

Halaman ini sengaja dikosongkan

LAMPIRAN A

Source Code

1. Ekstraksi dan merubah ke citra abu

```
function tri_frames = citragray()
global reader

for i = 1:3
    frame = step(reader);
    frame = im2uint8(frame);
    frame = rgb2gray(frame);
    tri_frames(:,:,i) = frame;
end
```

2. *Two Frame Difference*

```
function dual_diff_frames =
two_frame_diff(tri_frames, t)

global HEIGHT LENGTH

for i = 1:2
    dual_diff_frames(:,:,i) = ...
        abs(tri_frames(:,:,i + 1) -
tri_frames(:,:,i));
    dual_diff_frames(:,:,i) = ...

set_upper_bound(dual_diff_frames(:,:,i), t);
end

HEIGHT = size(dual_diff_frames, 1);
LENGTH = size(dual_diff_frames, 2);
```

3. Thresholding

```
function t = nilaithresh(tri_frames)

dtri_frames = double(tri_frames);

hmean = vision.Mean;
m = step(hmean, dtri_frames);

hstd2d = vision.StandardDeviation;
std = step(hstd2d, dtri_frames);

t = m - std;
```

4. Three Frame Difference

```
function tri_diff_frame =
tri_frame_diff(dual_diff_frames)

global HEIGHT LENGTH

for i = 1 : HEIGHT
    for ii = 1 : LENGTH
        if (dual_diff_frames(i, ii, 1) == 255
&& dual_diff_frames(i, ii, 2) == 255)
            tri_diff_frame(i, ii) = 255;
        else
            tri_diff_frame(i, ii) = 0;
        end
    end
end
end
```


5. Citra Dilasi

```
function dil_tri_diff_frame =
dilacitra(dual_diff_frames)

global HEIGHT LENGTH

se = strel('square',3);
dill_tri_diff_frame =
imdilate(dual_diff_frames(:,:,1), se);
dil2_tri_diff_frame =
imdilate(dual_diff_frames(:,:,2), se);

for i = 1 : HEIGHT
    for ii = 1: LENGTH
        if (dill_tri_diff_frame(i, ii) == 255
&& dil2_tri_diff_frame(i, ii) == 255)
            dil_tri_diff_frame(i, ii) = 255;
        else
            dil_tri_diff_frame(i, ii) = 0;
        end
    end
end
end
```

6. *Improved Three Frame Difference*

```
function im_tri_diff_frame =
im_tri_frame_diff(tri_diff_frame,
dil_tri_diff_frame)
global HEIGHT LENGTH

for i = 1 : HEIGHT
    for ii = 1 : LENGTH
        if (tri_diff_frame(i, ii) ~=
dil_tri_diff_frame(i, ii))
            im_tri_diff_frame(i, ii) = 255;
        else
```

```

                                im_tri_diff_frame(i, ii) = 0;
                                end
                                end
                                end
end

```

7. *Optical Flow* Horn-Schunck

```

function v = opticalflow(citfd)
global LENGTH k nz

opticFlow = opticalFlowHS('MaxIteration',
inf, 'VelocityDifference', 1);
flow = estimateFlow(opticFlow, citfd);
nz = nnz(flow.Magnitude);
fpm = sum(sum(flow.Magnitude));
v = fpm*k*3.6;

```

LAMPIRAN B DESAIN GUI

a. Halaman Utama



b. Pilih ROI



Halaman ini sengaja dikosongkan

BIODATA PENULIS



Penulis bernama Achmad Romli, lahir di Surabaya, 24 Maret 1992. Penulis merupakan anak bungsu dari 4 bersaudara dari pasangan Bapak Nawali dan Ibu Sarimah. Jenjang pendidikan formal yang ditempuh oleh penulis dimulai dari SDN Wonokusumo V/44 Surabaya (1999-2005), SMPN 27 Surabaya (2005-2008), SMAN 8 Surabaya (2008-2011). Setelah lulus dari SMAN 8 Surabaya, penulis melanjutkan studi ke jenjang S1 di

Departemen Matematika ITS pada tahun 2013-2017 melalui jalur SBMPTN dengan NRP 1213 100 105. Di Departemen Matematika ITS penulis mengambil Bidang Minat Matematika Ilmu Komputer. Selain aktif kuliah, penulis juga aktif berorganisasi di Himpunan Mahasiswa sebagai Staff Student Welfare HIMATIKA ITS (2015/2016), Staff Syi'ar Ibnu Muqlah (2014/2015), Staff Syi'ar Ibnu Muqlah (2015/2016) dan mengikuti beberapa kepanitian yang ada di dalam kampus. Jika ingin memberikan kritik dan saran, serta diskusi terkait Tugas Akhir ini, silahkan menghubungi email: aliansi.corp@gmail.com.